AD-753 543

Biocybernetic Factors in Human Perception and Memory

Stanford University

prepared for

Advanced Research Projects Agency

NOVEMBER 1972

Distributed By:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE

Semi-Annual Technical Report

(1 June 1972 to 30 November 1972)

TITLE OF WORK:	Biocybernetic Factors in Human Perception and Memory
CONTRACT NO:	DAHC15-72-C-0232
ARPA ORDER NO:	2190
PRINCIPAL INVESTIGATOR:	Dr. David C. Lai, (415) 321-2300, ext. 4544
NAME OF CONTRACTOR:	Stanford University
EFFECTIVE DATE OF CONTRACT:	1 June 1972
CONTRACT EXPIRATION DATE:	31 July 1973
AMOUNT OF CONTRACT:	$316,618.00

Sponsored By

Advanced Research Projects Agency

ARPA Order No. 2190

Biocybernetic Factors in Human Perception and Memory

SUMMARY

This project is concerned with the development of the biocybernetic concepts and techniques required for the analysis and training of skills useful for the manipulation and control of concrete memory functions. Our effort has been concentrated on those aspects of concrete memory which could be expected to make a substantial contribution to practical volitional memory, and we plan to bring them under stronger individual control. To realize the biocybernetic expansion of human memory, we choose to use the real-time computerized monitoring and feedback of spatial and temporal cues that serve as keys to memory retrieval. A PDP-15 computer system has been selected and purchased for this purpose. The selection of this computer system is based on its real-time capability. This technical report is concerned with the development of software and hardware for our purpose and the preliminary tests of some new techniques in real-time monitoring and prediction of brain states via real-time on-line analysis of electroencephalographic signals.

The PDP-15 computer system is an 18-bit computer with unique capabilities suited to high-speed data acquisition and processing. The system which we have purchased has a core memory of 16K words complemented with several processors and peripherals to handle our needs. The most significant feature is the RSX-15 (phase II) software system which makes the real-time monitoring possible. We have also developed special software and designed hardware for interfacing laboratory instruments. Details of these software/hardware combination are described in the report.

A new technique for real-time monitoring of brain states has been developed and partially implemented. This technique involves the realization of a discrete "phase-locked loop" system. The requirement of real-time processing has confronted us with many technical challenges. We have overcome some of them; others remain to be solved. These real-time monitoring and prediction scheme is described in detail in this report. Other methods such as the use of autoregressive processes for modeling have been investigated. It is also described in the report.

Preliminary works on eye-movement tracking have been completed. Extensive work in this area is to be followed. We give a somewhat detailed account of the preliminary eye-movement measurement scheme in the report.

In conclusion, our work has proceeded as planned in the proposal except that delays have been caused by the late delivery of the PDP-15. There is also the delay caused by diverting our effort to writing many utility programs which were thought to be supplied by Digital Equipment Corporation (DEC) through the misrepresentation of the manufacturer's representatives. These unforeseen delays have hindered our work and may upset the schedule for completing the work as proposed. In this report, we give an account of our technical accomplishments up to this point. The implications of this project for the Department of Defense lie in our specific mission of investigating the possibility of devising new and unusual techniques that will permit the development of stronger imagery in men with normal memories. Since it is definitely an asset for the military man to have the ability to absorb information rapidly and to retain it intact for long periods of time, we seek to develop and to intensify the post-stimulus imagery as far as possible in the direction of photographic memory.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

This research project is concerned with the development of the biocybernetic concepts and techniques required for the analysis and training of skills useful for the manipulation and control of the concrete memory functions, viz., those related to the more concrete sensory images. In particular, we concentrate on the problem of achieving biocybernetic expansion of human memory through the use of real-time computerized monitoring and feedback of spatial and temporal cues that serve as keys to memory retrieval. There is strong evidence which can be advanced to show that the human nervous system depends heavily upon spatial and temporal cues both in the encoding and decoding of memories especially those related to the sensory images.

Our scheme is first to develop and implement techniques for real-time monitoring and prediction of central nervous activities through the electroencephalographic signals. Then we shall use this information so as to arrange extraordinary coincidences between various brain states and the delivery of sensory stimulation. The visual stimuli are to be presented via the complex tachistoscopic batteries for either monopic or dichopic viewing. Meanwhile, the eye-position and eye-movements are to be measured by a unified system as to obtain their optical and electro-physiological estimators. Finally, combining these techniques and the use of a feedback scheme to close the control loop, we expect to obtain a tighter grip on the factors underlying image persistence and image dissipation. In this report, we describe our accomplishments in developing and implementing those techniques towards this goal.

We chose a PDP-15 computer system manufactured by Digital Equipment Corporation (DEC) for its unique capabilities suited to real-time data acquisition and processing to realize our schemes. Much time has been spent on bringing-up the computer system to its fully operational status. In the next section, we describe our effort in making the computer system function and fulfill our needs, and its integration with other laboratory instruments through our own designed interface hardware. In section III, we present a new technique for real-time monitoring and predicting various brain states via EEG signals and its partial implementation; we also give a method for modelling EEG signals and some of the results. The preliminary work on eye-movement measurement and tracking is presented in section IV. Some conclusion and observation are made in the last section.

The work has progressed as we proposed but not at the pace that we estimated. This is due to many unforeseen delays caused by the late delivery of the PDP-15 computer system and diversion of our efforts in writing many subroutines supposed to be supplied by the manufacturer.

II. PDP-15 COMPUTER SYSTEM AND ITS INTEGRATION WITH LABORATORY INSTRUMENTATION

The PDP-15 computer system which we have purchased includes the following items:

PDP-15/35 Disk Operating Advanced Monitor System, consisting of:
16,384 18-bit, 800-ns core memory
- LA30 DECWriter
- PC15 High Speed Paper Tape Reader/Punch
- KE15 Extended Arithmetic Element
- KA15 Automatic Priority Interrupt
- KW15 Real Time Clock

Advanced Moniotr System (cont.):
- TC15 DECTape Control
- TU56 Dual DECTAPE Transport
- RF15 DECdisk Control
- RS09 DECdisk Drive, 262,144 words

KM15 Memory Protect

KF15 Power Fail

FP15 Floating Point Processor

KT15 Memory Relocation

TC59D Magnetic Tape Transport Control for up to 8 TU10A, TU10B, TU30A, TU30b Magnetic Tape Transport Units

TU10A 9-Track, 45 ips Magnetic Tape Transport; 800 bpi

VP15C Oscilloscope Display, VR14 X-Y Display Unit (7" x 9" CRT) and Control

XY15AB 0.005-Inch Step 18,000 Steps/Minute 31-inch drum, Calcomp Model 563 and control

CR15 30 cpm Reader and Control (Punched Cards)

AA15B Multiplexer Control for up to 16 Type AAC3 12-bit digital-to-analog converter channels

AAc3 Digital-to-Analog Converter, single buffered, 0V to $\pm$ 10V

AD15 128 Channel A/D Converter (Medium Speed) Three-cycle data channel capacity with provision for mounting the first 32 channels; each 4-channel group requires one BA124

BA124 Four-Channel MOS FET Multiplex switch; one required for each 4-channel group

CSS 18-Bit Digital I/O Interface

RSX15 Software Package

RASP Software

LP15-F Line Printer (on lease)

This computer system is integrated with other laboratory instrumentation as shown by the block diagram in Figure 1. All of the interface hardware except the CSS-18 I/O interface are designed and built or to be built by us. We give a brief description of these interface hardware and a status report of their various stages of implementation.

(1) Interfacing of Special Peripheral Devices

(a) General: A general method for the interface of the special peripheral devices required for this project was developed using a specail device (the CSS 18 bit interface) provided by the computer manufacturer. This is a bidirectional interface package which does all of the address decoding and timing functions required of a device controller. Our use of several devices requires a device selector/data multiplexer be built, and then additional special conic (level shifting, inverting, etc.) unique to each device. Thus far, the special interface units have been designed, and, where noted, built. The device selector has not been completed. It is presently circumvented by paralleling several devices.

(b) Tachistoscope: The tachistoscope interface has been completed and is operational. The unique functions required were level shifting and pulse generation, for a total of six commands.

(c) Random Access Slide Projector: The interface unit has been designed and is currently under construction. Logic was required to allow parallel entry of data, and to properly time the control signals. Level shifting of all command and data lines is also required.

(d) Time-Code Generator: The interface for the tape search unit/ time code generator is not completed. The functions required are

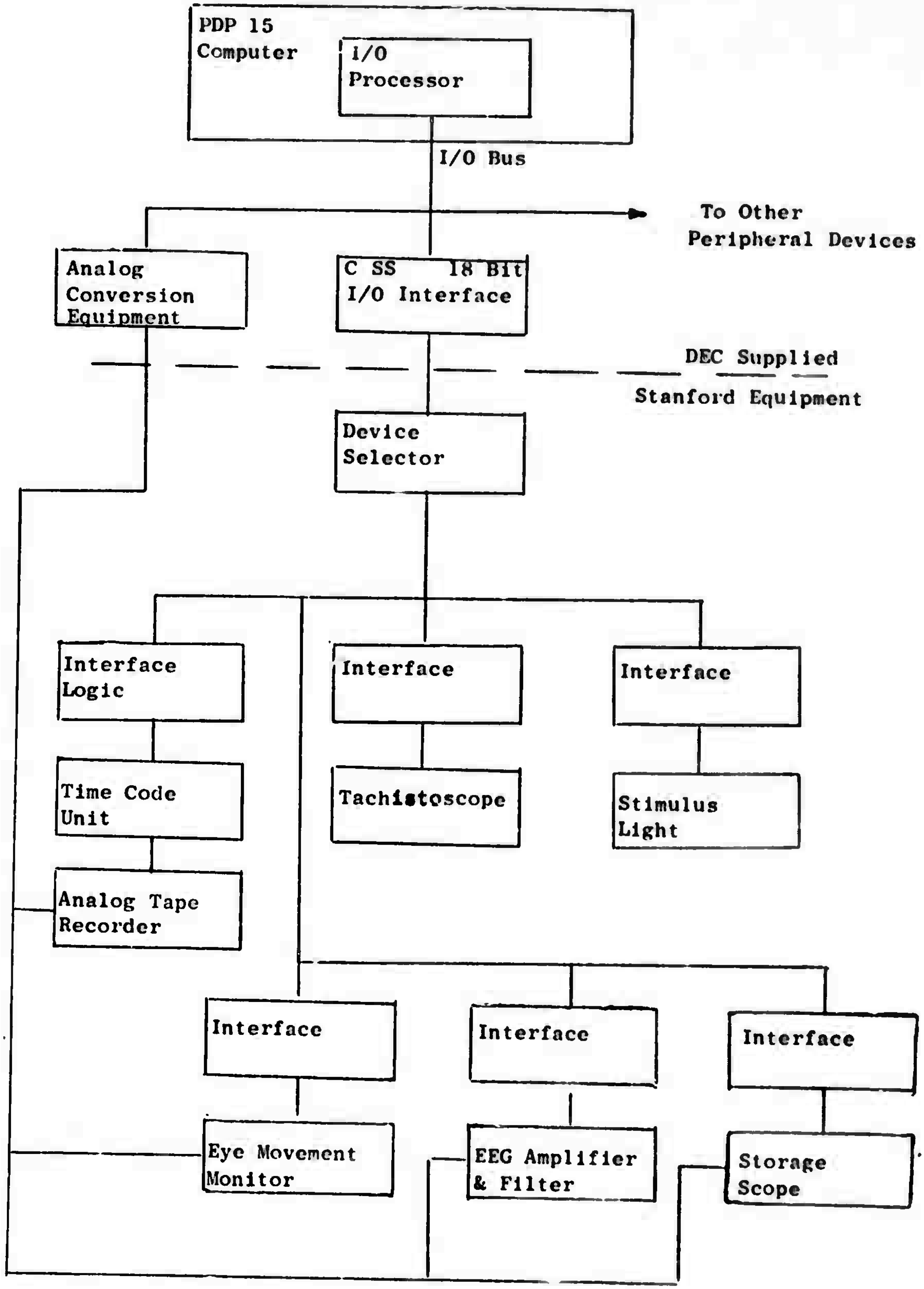

Fig. 1 PDP-15 Installation
Integrated With Other Laboratory Equipment

timing and control, and inversion of data. Final delivery of the TSU/TCG is not scheduled until later.

(e) Alden 3-Axis Recorder: The interface for this unit is not completed.

(2) Modification of Existing Peripheral Devices

(a) Analog-to-Digital Converter: A modification to this unit is in progress to add an external clock, to relieve the CPU of the timing workload

(b) Syst CRT: The present refresh type display unit is being replaced with a Tektronix type 611 storage display unit. This will allow use of the CRT during periods of high CPU utilization. Present plans are to retain the VR14 scope presently installed for off line tasks, driven by the system's digital-to-analog converters, and to use the storage unit as a display device for real time tasks.

To bring-up the PDP-15 computer to its operational stage and to fulfill our own special needs, we need to generate software which falls in the following three categories:

(a) General Support Programs and Subroutines: These include utility programs and library routines written for use by all project personnel. These are extensions of the operating system and utility packages produced by the manufacturer of the computer, which are required for day to day operations.

(b) Application Programs: These are programs which perform specific functions.

(c) Software investigations: Research effort required for production of functional programs. Most work is concerned with the timing constraints on real-time system operation.

Most of the software in categories (a) and (c) are completed since these form the basis for operation of the system in our application. In the sequel, we give a more specific account of what has been accomplished for each category.

(A) General Support Programs and Subroutines

The PDP-15 Real-Time Executive (RSX) was designed for a general scientific user. In order to tailor the system to our specific needs certain modifications to the DEC software were necessary. Unfortunately, these modifications could not be realized using the RSX system (due mainly to the large core requirements of the resident monitor). We were therefore forced to use DEC's disk oriented batch processing monitor (DOS) to update and assemble the RSX system.

The first such modification was extending the RSX teletype handler to simulate horizontal tabulation on our high-speed, teletype compatible teleprinter. Unlike a teletype, the LA30 has no hardware tabulation. Additional core was added within RSX to monitor the current position of the carriage, make note of new read/write requests, and print blank characters when necessary to achieve carriage movement corresponding to tabulation. We are now able to 1) enter columnar data with much less mental effort, and 2) reduce symbolic file space by at least a factor of two.

In order to allow the system to be "booted in" from disk without using the paper tape reader, we made a second extension to RSX. The system must be "booted-in" to start the system in the morning and to refresh memory following a severe system failure. The delivered system required a "warm start" tape to be placed in the paper tape reader along with the setting of certain console switches. By placing a copy of the warm start program within the resident monitor (less than 20 words) we are now able to warm start the system by simply depressing the START switch. Since paper tape is cumbersome to work with, we save time and effort.

In addition to a powerful operating system most computer users need an assortment of utility routines such as memory and magnetic tape dumps, graphics support (plotter, CRT), special purpose subroutines, magnetic tape copy, etc. Although the DOS system supplies some utility software, RSX includes virtually none. We describe in detail the current contents of our utility package generated by us.

CORE/DISK DUMP

An RSX user may examine the contents of the disk or memory through use of the MCR function, OPEN. OPEN allows the inspection of only one word per invocation; however, it is unsuited for displaying large, contiguous blocks of core and disk. For this reason an additional MCR function, DMP, was written. The syntax definition of the DMP command is as follows:

DMP␣[D] from [␣to]

where from = starting core/disk address

to = ending core/disk address

from ≤ to.

Both the starting and ending addresses must be octal constants. The optional parameter, "D", is specified when a disk dump rather than a core dump is required. The dump lisitng will be written on logical unit LUN 16 which is normally assigned to the line printer. DMP first extracts the starting and ending addresses from the command line, and then initializes itself by filling the output buffer with blanks. As each word is received from core or disk, it is placed into the output buffer. When the buffer is full (eight words), it is printed on LUN 16. This procedure continues until the requested area is dumped. To understand the making of the program is through the study of its flow-chart. The flowchart for DMP is shown in Figure 2.

Magnetic Tape Dump

A prgoram (DTDUMP) is available to DUMP in octal, selected blocks of a DEC taped assigned to LUN 19. Since DTDUMP is a task rather than an MCR function, its execution is initiated through the REQUEST function of RSX. The user is asked to specify the beginning and ending blocks to be dumped after which a listing is produced on LUN 16 (line printer). The program re-cycles until a starting block of zero is given in response to DTDUMP's question.

MTDUMP Program

This is a system utility program written to print out the data on 9-track 800 BPI industry compatible magnetic tape, in an octal format. It is a task, called by the system request function. The program uses the console teletype for user input of the expected record size and the number of records. Each record is read in, the format converted, and dumped to the printer. Printing suppression is used for all zero lines. The program will run even if the incorrect record size is specified, with the extraneous values suppressed. Termination occurs on end of

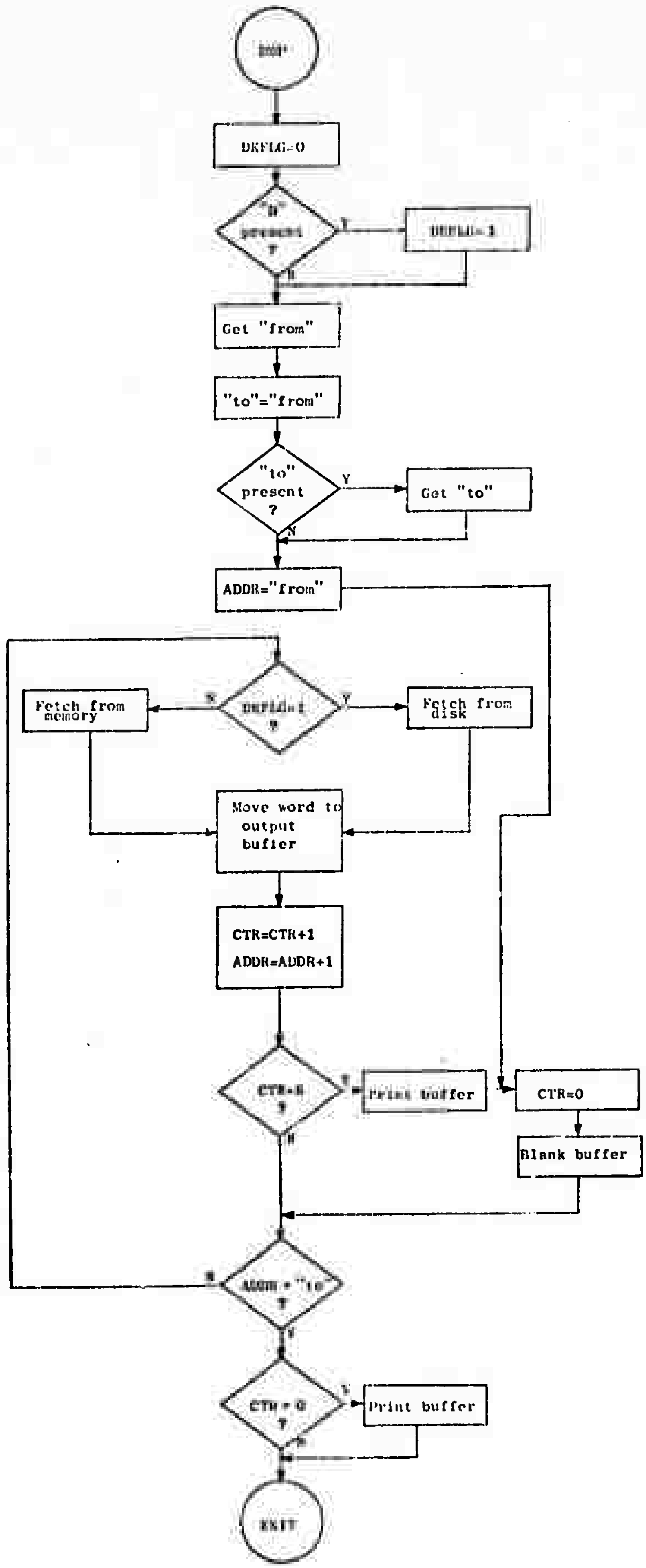

Fig. 2 Flowchart for Core/Disk DUMP Program

file, or completion of the required number of records. No rewind is built in to the program, to allow restarting the dump after a known number of records, or after some significant event on the tape determined by a preceding program. A brief flowchart is shwon in Figure 3.

Graphics Software

The RSX software system does not include routines to support the CalComp incremental X-Y plotter. A temporary subroutine (PLOT) was written to fulfil our present needs. This routine is called out by the following FORTRAN statement:

CAL PLOT (N, IDATA ,ICHAR)

where N = number of points to be plotted

$1 \le N \le 10$

IDATA = an INTEGER array of ordinate values

$0 \le IDATA(I) \le 2000$

ICHAR = an INTEGER array of character codes

$0 \le ICHAR(I) \le 3$

PLOT was designed to display time-varying functions with a constant time interval between data points. For this reason no abscissa values need be specified in the subroutine call. As noted above the ordinate values, IDATA(I), must be normalized to lie between zero and 2000 inclusive. Data points outside of this range are simply ignored. Each call to PLOT advances the paper one time unit in the positive "X" direction.

To each data value in IDATA is associated a corresponding character code in ICHAR. It is capable of drawing four distinct symbols for any IDATA(I) as shown in the following table:

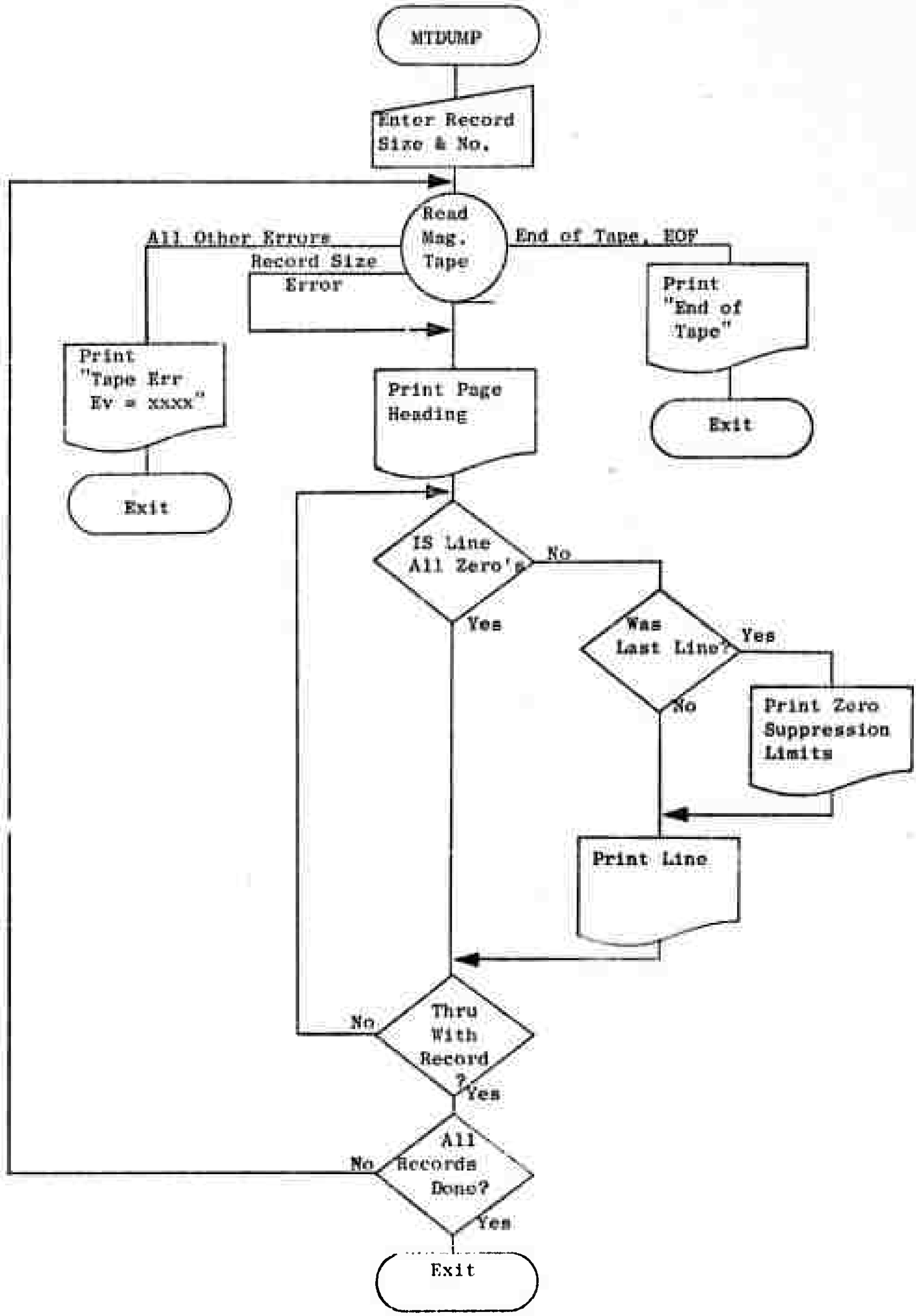

Fig. 3 Flowchart for MTDUMP Program

Character Code	Symbol	Description
0	|	Vertical Line
1	—	Horizontal Line
2	□	Square
3	+	Plus Sign

ICHAR allows the programmer to use separate symbols for each graph of a multiple graph plot. If ICHAR is not specified, PLOT will draw a vertical line for all IDATA values.

A flowchart for PLOT is shown in Figure 4. After moving the two input arrays into a buffer, PLOT sorts IDATA (and corresponding elements in ICHAR) in order to minimize the time required to plot the "N" points. The points are then plotted in their sorted order. Notice that an error conditions result in an immediate exit from PLOT. A more sophisticated graphic plot routine is being written.

<u>Special Purpose Subroutines</u>

The PDP-15 has a group of eighteen "data switches" or "sense switches" (numbered 0,1,...,17) mounted on the operators console. Since the positions of these switches may be dynamically determined through the execution of a hardware instruction (OAS), programs may use these switches for low volume data input. The standard DEC software allows only assembly language programmers to determine the data switch settings. In order to extend this capability for FORTRAN programmers, two FORTRAN callable routines have been created.

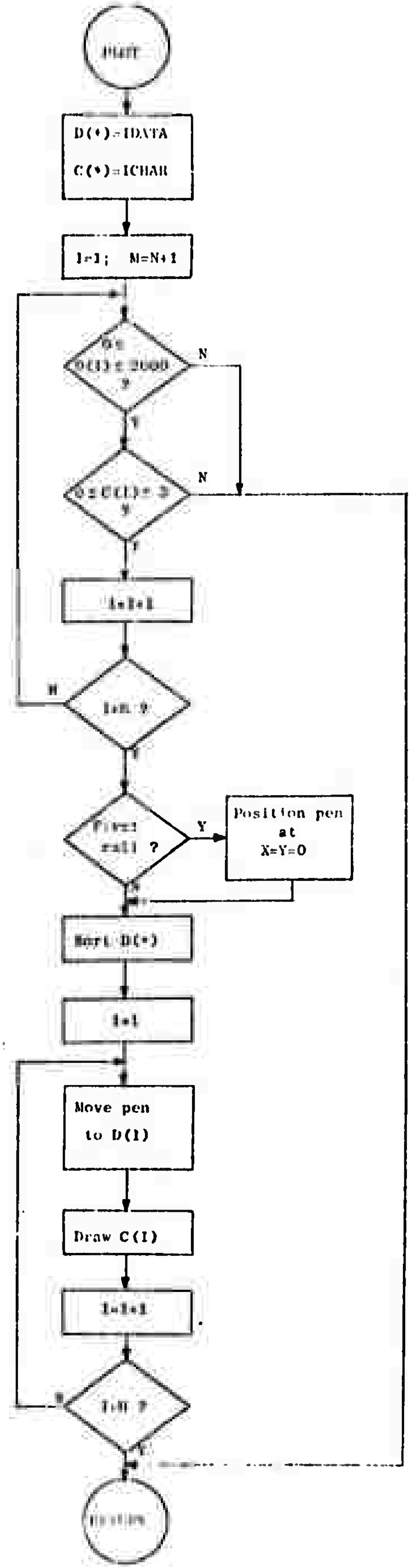

Fig. 4 Flowchart for PLOT Program

The eighteen data switches may be thought of as single eithteen-bit binary integer since each switch must be in only one of two possible positions. When a user wants to ascertain the settings of all or most of the switches, one may include the INTEGER FUNCTION LAS in his program. When calling LAS an argument must be specified. For example, the following short FORTRAN program will print the current status of the data switches as a six-digit octal integer:

```
          N = LAS(0)
          PRINT (13,1) N
1         FORMAT (IX,Ø6)
          END
```

Frequently a program contains a number of optional features that the user must selectively request. A simple way to tell this type of program what options are needed is to assign one data switch per option. In order to allow FORTRAN programs to test single data switches, a LOGICAL FUNCTION ISWTCH was written. This routine is referenced as follows:

ISWTCH(I)   where I = INTFGER variable or constant
$(0 \leq I \leq 17)$

ISWTCH(I) returns a "TRUE" value if and only if data switch "I" is on. A FORTRAN statement that would pause whenever both data switches #0 and #6 were on is coded as

```
IF (ISWTCH(0)  .AND.  ISWTCH(6))  PAUSE.
```

A flowchart depicting ISWTCH's operation is given in Figure 5.

<u>RIPOFF Subroutine</u>

This subroutine was written to convert data recorded on magnetic tape, or analog-to-digital converted data to a format consistent with

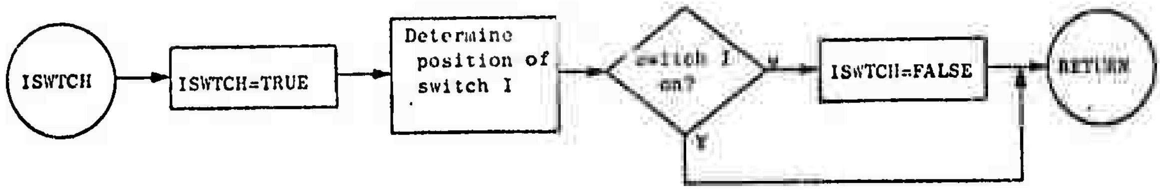

Fig. 5 Flowchart for ISWITCH Program

the internal formats for data values. Certain magnetic tape input/output routines pass the data to the calling program without removing the parity bits. Also, A/D converter data is 12-bits wide. Thus negative values are not properly represented in an 18-bit word. This subroutine corrects for both problems. It is currently in use as a FORTRAN callable library routine.

REA Subroutine

The RSX operating system utilizes core partitions outside of the monitor for input-output device handlers. The use of these partitions is not allowed by other tasks if the device is on line and in use by the system. This subroutine was developed to dynamically reallocate the core space used by these handlers by reassigning the device off line. This allows the use of a peripheral device (such as a line printer) to be deferred during real time operations (such as digitizing), then returned for use during a following off-line operation, under program control. It is a FORTRAN callable routine, currently used in the Digit I program.

GT132 Subroutine

One other special purpose subroutnine is included in the disk-resident user library. The routine was created because of an early need to read dititized EEG magnetic tapes. Normally these tapes would have been read by the standard FORTRAN input statements. However, at the time these tapes were needed a FORTRAN program could not read them because of errors in the hardware and/or software.

Therefore, a subroutine, GT132, was written which would read one record (132 characters long) from a digitized EEG tape and pass this record to the user. The calling sequence for GT132 is

```
Call GT132(BUF,EOF)

        where  BUF = 27 element DOUBLE INTEGER
                     array
               EOF = FORTRAN statement number
```

Each call to GT132 places a tape record into BUF unless an end-of-file is encountered. In the latter case GT132 returns control to the statement specified in EOF instead of executing a normal RETURN statement.

As shown in the flowchart in Figure 6, GT132 translates the EBCDIC characters on tape into the ASCII character equivalents required by RSX after first reading the tape in binary. DEC just recently corrected the problem so that we are now able to read the tapes with the FORTRAN READ statement.

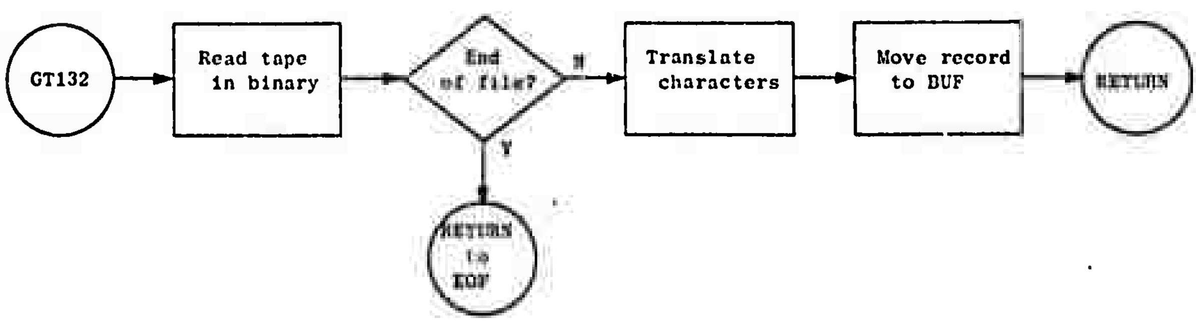

Fig. 6 Flowchart for GT132 Program

(B) Applications Programs

Digit 1

This program was developed as a utility program to digitize analog data (online or from analog tape) and record it on digital magnetic tape for offline analysis. The present version is an intermediate program in two ways. First, it is expected to be revised to take advantage of modifications in the analog to digital conversion hardware to allow faster sampling rates. Second, it is intended for general purpose data acquisition until command and control functions are being issued real time. At that point, no longer will all data be recorded and then analyzed later; only selected data will be output as the real time function is being performed.

The program uses a sequence of several tasks to complete the data collection and recording. Digit 1 is requested by the user; it performs initialization and reallocates a partition for use by the actual data collection routine. Data for an identification file is input either from the card reader or from the teletype, and this header file is written on the output tape. When complete, the data acquisition task (INTK5) is fixed in core and scheduled to run at 5 millisecond intervals. This task inputs data via the A/D converter, and outputs to magnetic tape, using buffered output. Following completion of data collection (as indicated by the user setting a console switch), task digit 2 is run. This task cancels and removes INTK5 from core, and allocated the partition back to the line printer handler. A brief tape verification is then run, printing the header record and scanning for record numbering errors as an indication of data loss. A general flowchart depicted in Figure 7 shows program operation.

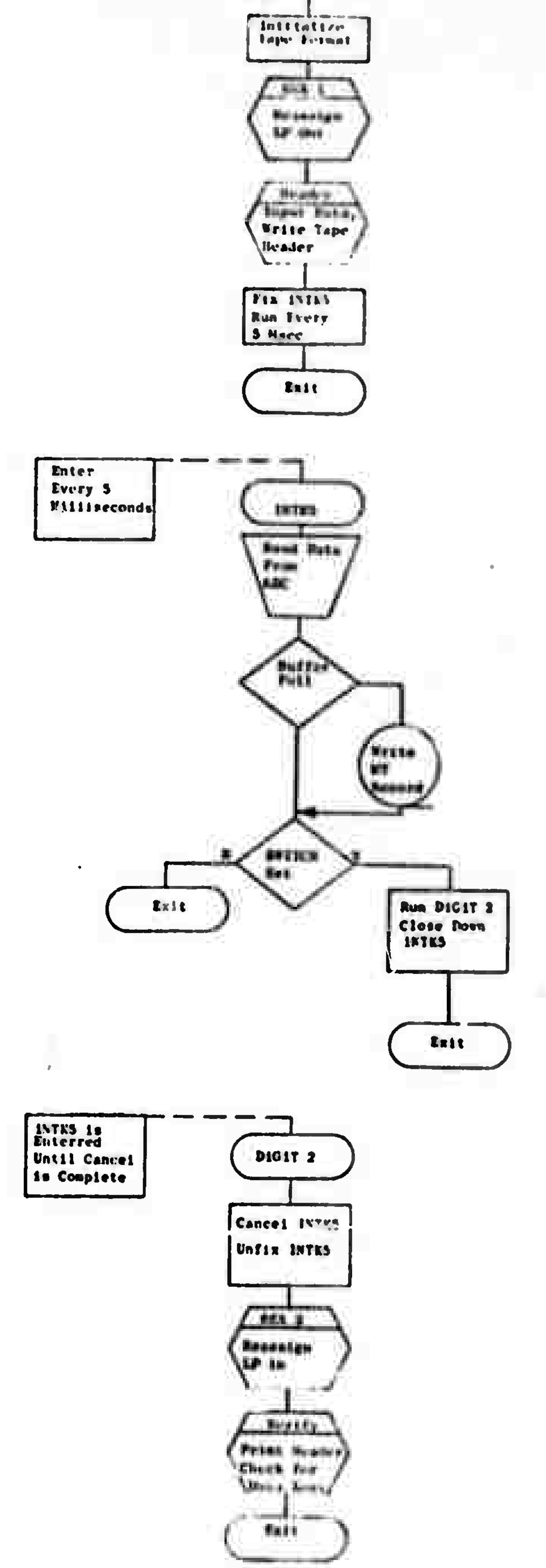

Fig. 7 Flowchart for DIGIT 1 Program

CRTPCT, SYSTIM

These programs were a derivative of the digitizing routine written to output data values to the system CRT During data acquisition. A single circular buffer of data was filled during A/D sampling for CRT output, and the buffer was repetitively scanned. Any of the four input channels can be viewed, with alphnumeric identification of the channel. System time is also displayed. The program is used primarily as a system checkout aid, since the CPU time required for rescan of the CRT degrades timing accuracy. The program is being converted to run with the storage CRT, and better results are expected.

(C) Software Investigation:

Practically all of our data are in analog form. It is imperative that we should have, in operation, a good analog-to-digital converter used to input data. An I/O handler package, produced by the manufacturerer, was available, and several versions of INTK were written to take advantage of this handler. A measurement method was designed to utilize a square wave input (at 10HZ) and an offline data analysis program which searched through the data to determine the consistency and accuracy of A/D operation. It became obvious that accuracy of the A/D was not a significant cause of error, but that millisecond level scheduling of the data collection task was causing severe CPU utilization problems. The strategy chosen was to tune both the hardware and the software for optimum A/D operation.

Accordingly, the standard I/O handler was replaced with a high speed, less general driver which reduced the need to schedule two tasks virtually concurrently. In addition, a hardware modification (not yet implemented) was designed to allow the A/D to operate on its own clock rather than requiring a CPU driven task to schedule each A/D operation. It is expected that in combination, these will allow a substantial increase in the digitizing rate that can be supported by the system.

The present version of the digitizing program, digit 1, utilizes the high speed driver described above. The results of the analysis were that the maximum error rate under the present configuration approaches one sample lost due to high CPU loading every two seconds.

Problems of this type are not analytically solved. The solutions are found with the particular operating system and hardware involved. Further work of this type is expected as more tasks operate in a real-time environment. A significant effort in the systems operation is to eliminate scheduling problems, such as this one.

III. REAL-TIME MONITORING TECHNIQUE AND MODELLING OF EEG SIGNALS

We describe here a new technique for real-time monitoring and predicting brain states via EEG signals. The EEG waveform is assumed to be a narrow-band process whos spectrum is centered around a mean frequency which may be slowly varying. Designating the EEG signal by f(t), we have $f(t) = e(t) \cos\left[u(t)\right] = e(t) \cos \omega_o t + \varphi(t) =$

$$s(t) \sin \omega_o t + c(t) \cos \omega_o t \quad ;$$

By tracking the central frequency ω_o , we may represent the signal by the two processes $s(t)$ and $c(t)$. If the process is stationary whose spectrum is symmetric about ω_o , then $s(t)$ and $c(t)$ are uncorrelated so the prediction would be easier to deal with.

To obtain the in-phase component $c(t)$ and the quadrature component $s(t)$, we use a discrete "phase-locked loop" system as described in the block diagram shown in Figure 8. This system tracks the center frequency ω_o and resolves the EEG waveform into $c(t)$ and $s(t)$. Once $c(t)$ and $s(t)$ are obtained for a window width of the EEG signal, we reduce the data by taking the mean value, the mean slope, etc. This is done by fitting a linear segment to the data in the window through recursive filtering. The values thus obtained serve as data vector components.

The key to the sucess of this real-time monitoring technique is the discrete phase-locked loop system depicted in Figure 8. This system includes:

(a) Voltage-controlled oscillator

(b) Frequency discriminator

(c) Low-pass filters

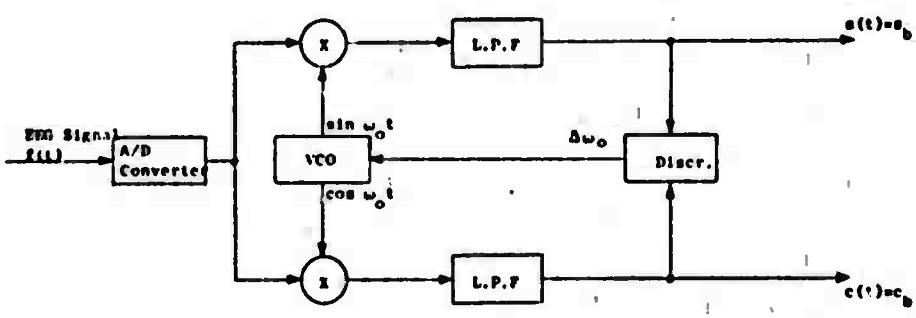

Fig. 8 A Discrete Phase-Locked Loop System

They are realized as follows:

(a) The VCO is realized by generating the sine and cosine waveforms through the difference equations

$$S(N) = \alpha S(N-1) + \beta C(N-1)$$

and $$C(N) = \alpha C(N-1) - \beta S(N-1) \quad .$$

Since the frequency and the amplitude are functions of α and β, we control these parameters in such a way as to maintain a constant amplitude and to change the frequency in accordance with the output of the discriminator. Specifically, we maintain unit amplitude by making $\alpha^2 + \beta^2 \simeq 1$ and increase or decrease the frequency by increasing or decreasing the parameter β according to the output of the discriminator.

(b) We have not yet found a satisfactory frequency discriminator which operates on sampled data. The one which we have tried so far is of the slope type, i.e., the frequency is related to an averaged slope of the waveform. It seems that we have to learn more statistical properties of the rate of change of the center frequency of the EEG α - waveform before we could decide on a particular type of frequency discriminator. This rate of change of frequency could be measured directly from EEG signals [1].

(c) The lowpass filters are a standard 3-pole Butterworth filter with a 3-db bandwidth about 5 Hz. This digital filter is discussed in [2]. The two output signals c(t) and s(t) of the previous system are then fed seperately into linear digital filters that fit to each of them the best linear approximation in a window of k samples in the least mean square sense. The width of the window was chosen to be 0.15 sec. The two parameters of the linear approximation, viz., the mean and the slope, are calculated recursively by digital filters known as the "comb" and "slope" filters. These filters are the digital realization of the transfer functions given by

$$c(j\ ) = \frac{\sin Kx}{\sin x}$$

$$s(j\ ) = \frac{\sin\left[(K+1)x\right]}{4\sin^2 x} - \frac{(K+1)\cos Kx}{4\sin x}$$

where $c(j\ )$ and $s(j\ )$ are the amplitude response of the comb and slope filters, respectively; and x is the normalized frequency. The value for K is 63 which corresponds approximately to the window width of 0.15 sec. The normalized frequency X is calculated by

$$X = \frac{\omega T}{2}$$

where T is the sampling period (2.5 msec), ω is measured by radians/sec.

The phase characteristics of the two filters are of the same linear type

$$\text{phase}(W) = \frac{(K-1)}{2} WT \quad .$$

These filters are block diagrammed in Figure 9.

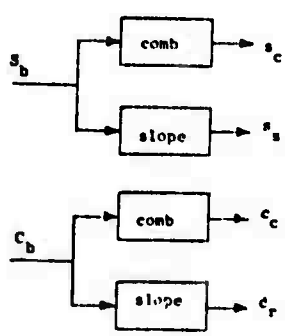

Fig. 9 Block Diagram for Linear Fitting Filters

It should be noted that the four outputs of these filters contain information about the frequency and may be used as frequency discriminator. If higher order "derivatives" are needed, they can be obtained by using recursive filters in a similar fashion.

We have not yet implemented a prediction scheme; however, a scheme to be described has been developed. Let X^k be a vector whose components consist of K past samples of EEG process, i.e.,

$$X^K = \begin{bmatrix} X_N \\ X_{N-1} \\ \vdots \\ X_{N-K+1} \end{bmatrix}$$

where N represents the present instant of time. Our aim is to predict the value of the process sample at the future instant L, viz., X_{n+L}. Because of the real-time prediction requirement, we have to transform X^K into another vector Y^M with

$$Y^M = QX^K$$

where Q is an MXK matrix with $M<<K$. Our prediction algorithm will operate on Y^M rather than X^K to produce the predicted value of X_{N+L}. For instance, the predicted value $\hat{X}_{N+L}$ would be obtained by

$$\hat{X}_{N+L} = (B^M, Y^M) = \cdot \sum_{i=1}^{M} B_i Y_i \quad ,$$

if a linear predictor is used. We are in the process of finding the optimal B^M and Q for a given L,M, and K. These optimal B^M and Q are time-varying for non-stationary processes such as the EEG process. In this situation, they will have to be tracked by using measurements of the prediction error or by some other means.

In closing the control loop, we need an adaptive model whose output is used as an input to the controller. This adaptive model serves as a check for our outputs from the predictor. An autoregressive process has been used for the modelling of EEG. In the following, we describe this method and some results obtained so far.

An autoregressive (AR) process of order p can be described by the equation

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + e_t$$

where x_t , x_{t-1},..., are the values of the process at equally spaced times $t, t-1, \ldots$, and the e_t are zero mean, independent, identically distributed random variables, i.e. white noise. Here it is assumed that the mean μ of the process is zero. Otherwise x_t should be replaced by $\tilde{x}(t) = x(t) - \mu$ in the above equation.

An AR process can be stationary or non-stationary. It can be shown that for the process to be stationary, the roots of the characteristic equation

$$\phi(B) = 1-\phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p = 0$$

must lie outside the unit circle [3]. For a stationary AR process the autocorrelation function consists of a mixture of damped exponentials and damped sinusoids, depending on the roots of the above equation. It should be noted that the autocorrelation function of an EEG exhibits a damped sinusoidal behaviour [4].

The above model contains p+1 parameters ϕ_1, ϕ_2, ..., ϕ_p and the variance of the noise. Since an EEG is a varying amplitude, frequency and phase process, it is estimated that at least a fourth order process will be needed to model it [5].

We estimate the parameters by using the following method:

Consider the equation

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \dots + \phi_p x_{t-p} + e_t$$

multiply by x_{t-k} and take expectations to get

$$c_k = \phi_1 c_{k-1} + \phi_2 c_{k-2} + \dots + \phi_p c_{k-p} \qquad k > 0$$

where c_k is the autocovariance at lag k and $E(x_{t-k} \cdot e_t) = 0$, since the noise being white, is uncorrelated with the past values.

Dividing by the variance c_o,

$$r_k = \phi_1 r_{k-1} + \phi_2 r_{k-2} + \dots + \phi_p r_{k-p} \qquad k > 0$$

where r_k is the autocorrelation at lag k.

Substituting $k = 1,2,\dots p$ in the above difference equation,

$$r_1 = \phi_1 + \phi_2 r_1 + \dots + \phi_p r_{p-1}$$

$$r_2 = \phi_1 r_1 + \phi_2 + \dots + \phi_p r_{p-2}$$

$$\vdots$$

$$r_p = \phi_1 r_{p-1} + \phi_2 r_{p-1} + \dots + \phi_p$$

which are the Yule-Walker equations [6], [7]. If the actual autocorrelations were known, the parameters $\phi_1, \dots, \phi_p$ can be obtained by solving these linear equations. The <u>Yule-Walker estimates</u> are obtained by using the estimated autocorrelations $\hat{r}_k$ in the above equations.

For $k = 0$, $E(x_t . e_t) = E(e_t^2) = \sigma_e^2$ so that

$$c_o = \phi_1 c_1 + \phi_2 c_2 + \ldots + \phi_p c_p + \sigma_e^2 .$$

and σ_e^2 can be estimated by substituting the estimated c's and ϕ's .

The Yule-Walker estimates are fairly accurate. However, the least squares and the maximum likelihood estimates are statistically more significant. If we assume that the process is Gaussian (i.e., the e_t are normally distributed), then it can be shown [3] that the least squares estimates are given by

$$D_{12} = \hat{\phi}_1 D_{22} + \hat{\phi}_2 D_{23} + \ldots + \hat{\phi}_p D_{2,p+1}$$

$$D_{13} = \hat{\phi}_1 D_{23} + \hat{\phi}_2 D_{33} + \ldots + \hat{\phi}_p D_{3,p+1}$$

$$\vdots$$

$$D_{1,p+1} = \hat{\phi}_1 D_{2,p+1} + \hat{\phi}_2 D_{3,p+1} + \ldots + \hat{\phi}_p D_{p+1,p+1}$$

where $D_{ij} = D_{ji} = x_i x_j + x_{i+1} x_{j+1} + \ldots + x_{n+1-j} x_{n+1-i}$ (n is the total number of terms in the series). The least-squares estimates can then be obtained by solving the above equations.

The exact maximum likelihood estimates can only be obtained by solving certain nonlinear equations. However, approximate estimates are obtained by solving certain nonlinear equations. However, approximate estimates are obtained by sloving linear equations similar to the above set but with D_{ij} replaced by $D^*_{ij} = \frac{nD_{ij}}{(n-i-j+2)}$. For large n , if the estimates are not close to the boundary between the stationary and non-stationary zones, the differences between three estimates is small.

The model is then used to predict the EEG process. These predicted values are used to check the predicted values from raw EEG. The prediction scheme by the use of the model is described next. The minimum mean-square error prediction filter is simply described by the difference equation

$$\hat{x}_t = \phi_1 \hat{x}_{t-1} + \phi_2 \hat{x}_{t-2} + \cdots + \phi_p \hat{x}_{t-p}$$

so that the prediction algorithm is quite simple. It can be seen that the characteristic equation of this filter is

$$1 - \phi_1 Z^{-1} - \phi_2 Z^{-2} - \ \ldots \ - \phi_p Z^{-p} = 0$$

so that the filter is stable if the roots of the above equation lie inside the unit circle. Thus the filter is stable if and only if the parameters correspond to a stationary process. Once the parameters have been estimated, the model has to be checked by using goodness-of-fit tests. An estimate of e_t is

$$\hat{e}_t = x_t - \hat{\phi}_1 x_{t-1} - \hat{\phi}_2 x_{t-2} - \ldots - \hat{\phi}_p x_{t-p}$$

It can be shown that [3]

$$\hat{e}_t = e_t + 0\left[\frac{1}{\sqrt{n}}\right]$$

so that as n increases, $\hat{e}_t$ approximates white noise more and more. The estimated autocorrelations of e_t can be shown [8] to be distributed approximately normally with mean zero and variance $\frac{1}{n}$. Then $\hat{e}_t$ can be calculated and its autocorrelations estimated, which can be used to check the model.

Another test is to use

$$Q = n \sum_{k=1}^{k} r_k^{\,2}(e)$$

where the r_k's are the estimated autocorrelations of $\hat{e}$. If the model is adequate, Q is approximately distributed as $\chi^2(K-p)$ [9]. This can be used to get a quantitative result about the goodness of fit.

Now, we will give some of the results obtained. We used the EEG data provided by NASA-Ames Research Center. Two types of data were used, one a spontaneous EEG and the other an EEG obtained when stroboscopic light was flashed into the closed eyes of the subject at the approximate alpha frequency. The data was sampled at 833.3 samples per second.

The processing of the data was carried out on a PDP-15. The data was first filtered using a narrow band-pass digital filter so as to isolate the alpha-frequency component. The filtered data was then divided into small blocks covering a time interval of about 1.2 seconds each, over which the EEG process may be regarded as stationary. A computer program which gives all three types of estimates, computes $\hat{e}_t$ and its autocorrelations and also computes Q, was used. Some approximate methods can be used to estimate the order p of the process but it was found simpler and better to specify it as a variable which could be input to the program. In the actual program, p could be varied between 2 and 10.

The results obtained indicate that an AR process model does not fit the EEG. The estimated autocorrelation of $\hat{e}_t$ were quite high and the χ^2 test also indicated a lack of fit.

The predictive filter was used and the actual and predicted time series was plotted on a calcomp plotter. It was found that for some least squares and maximum likelihood estimates, the predictive filter is unstable which indicates that the maximisation occurs in the nonstationary region.

The results obtained so far do not live up to our expectation. The model may have to be refined. Other models are being investigated.

IV. PRELIMINARY WORK ON EYE-MOVEMENT TRACKING

All of the equipment necessary for the initial eye-movement experiments has arrived and software for processing the eye-movement data is under development. We have become familiar with adjusting the sensors on the eye-movement apparatus to obtain large and symmetrical deflections in both horizontal and vertical directions. These adjustments are somewhat involved and difficult, but they are very necessary for obtaining accurate results. With practice we have been able to get better and better adjustments, and we are now satisfied with our results and confident of our ability to rapidly and accurately adjust the sensors for optimum performance.

We have nearly completed the calibration procedures and algorithms for the eye-movement measuring instrument. These calibrations are necessary for obtaining accurate measurements because they provide a calibration matrix as a reference for all subsequent experimental measurements. Consequently, we are taking great care to insure a valid calibration procedure that is also convenient to use, because it will probably be necessary to check our calibrations frequently throughout the course of our experiments.

Basically, the calibration procedure involves the following:

(a) digitizing the subject's eye movements on-line with the A/D converter and displaying these as points on the computer oscilloscope, much the same as displaying the movements directly on an oscilloscope. This computer "feed through" is used so that the experimenter can tell the computer to store certain calibration points for later reference during the actual experiment; (b) the experimenter instructs the subject to fixate on particular points of a calibration matrix (both square and circular matrices are used) and adjusts the eye movement sensors so as to get the scope tracing to approximate as closely as possible the particular matrix being observed; (c) when a good approximation has been obtained on the scope, the experimenter instructs the subject to look at a particular point in the matrix and then tells the computer via the switch register or the teletype to store that point's coordinates in a matrix array. This is an averaging process which samples the EMM output over a period of time while the subject is fixating on the given point so that the effect of eye blinks and other artifacts can be reduced or ignored; (d) when all the calibration points have been inputted, the experimenter can display the points to check that the matrix was inputted properly; if not, (c) and perhaps even (b) should be repeated until a good calibration is obtained.

After completing the calibration routines, we will start on algorithms for processing data continuously and for doing things such as plotting the eye movements on the plotter (not real-time plots of course). These routines will depend heavily on the calibrations described above, as the calibration points will be used as references for interpolating the positions of the movement data.

V. CONCLUSION

The work has been proceeding along the lines as planned though not at the pace as expected. We encounterd some difficulties in the beginning of this project. These have largely been overcome. The PDP-15 computer system is running satisfactorily. There still remain several laboratory instruments to be interfaced with the computer system and some utility subroutines to be written. In the next period, we will concentrate on generating application software packages and realizing algorithms for such schemes as the real-time monitoring and prediction.

We have investigated some real-time monitoring and prediction schemes and some modelling techniques. One of such a scheme has been partially implemented and tested. A modelling technique has been implemented and tried out. The preliminary results obtained from the model indicate that the model has to be refined. Also, preliminary work on the eye-movement monitoring and tracking has been completed. Further work on the measurement of eye-movements and eye-positions are in progress.

Our immediate work is to acquire experimental data. With appropriate experiments and the analysis of the eye-movement and cortical-state data, we can determine the right timing and sequencing of saturation and test stimuli for obtaining persistent and exclusively positive after-images or persistent and exclusively negative after-images. By using the continuously updated information concerning the eye-position and cortical state for adjusting the stimulus parameters, we will be able to enhance the vividness and persistence of these after-images. We then will close the control loop for enhancing memory functions by incorporating all these information and techniques implemented through the system as block diagramed in Figure 1. The work remaining to be accomplished is the integration of the various techniques, implementation of various schemes, and the determination of the appropriate stimulus parameters via the analysis of the experimental data.

REFERENCES

1. Lai, D. C. and Lux, R. L., "Application of Frequency Discrimination Technique to the Analysis of Electroencephalographic Signals" Proc. 1972 National Electronics Conference, 27: 80-85, Oct. 1972.

2. Gold, B. and Rader, C.M., Digital Processing of Signals, New York; McGraw-Hill, 1969.

3. Box, G. E. P. and Jenkins, G. M., Time Series Analysis Forecasting and Control, San Francisco; Holden-Day, 1970.

4. Prabhu, K. P. S., "On Feature Reduction with Application to Electroencephalograms", Tech. Rep. 615, Division of Engineering and Applied Physics, Harvard Univ., Cambridge, Mass.

5. Anderson, T. W., The Statistical Analysis of Time Series, New York John Wiley, 1971.

6. Yule, G. U., "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wölfer's Sunspot Numbers", Phil. Trans., A225,267, 1927.

7. Walker, G., "On Periodicity in Series of Related Terms", Proc. Royal Soc., A131,518, 1931.

8. Anderson, R. L., "Distribution of the Serial Correlation Coefficient", Ann. Math. Stat., 13,1, 1942.

9. Box, G. E. P. and Pierce, D. A., "Distribution of Residual Autocorrelations in autoregressive-integrated Moving Average Time Series Models", Jour. Amer. Stat. Assoc., 64, 1970.